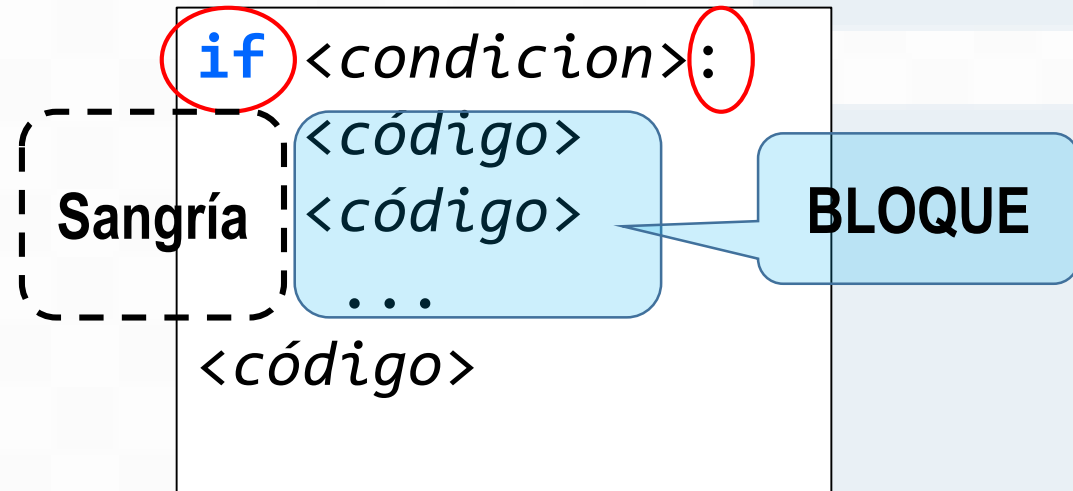


REPASO CONDICIONALES



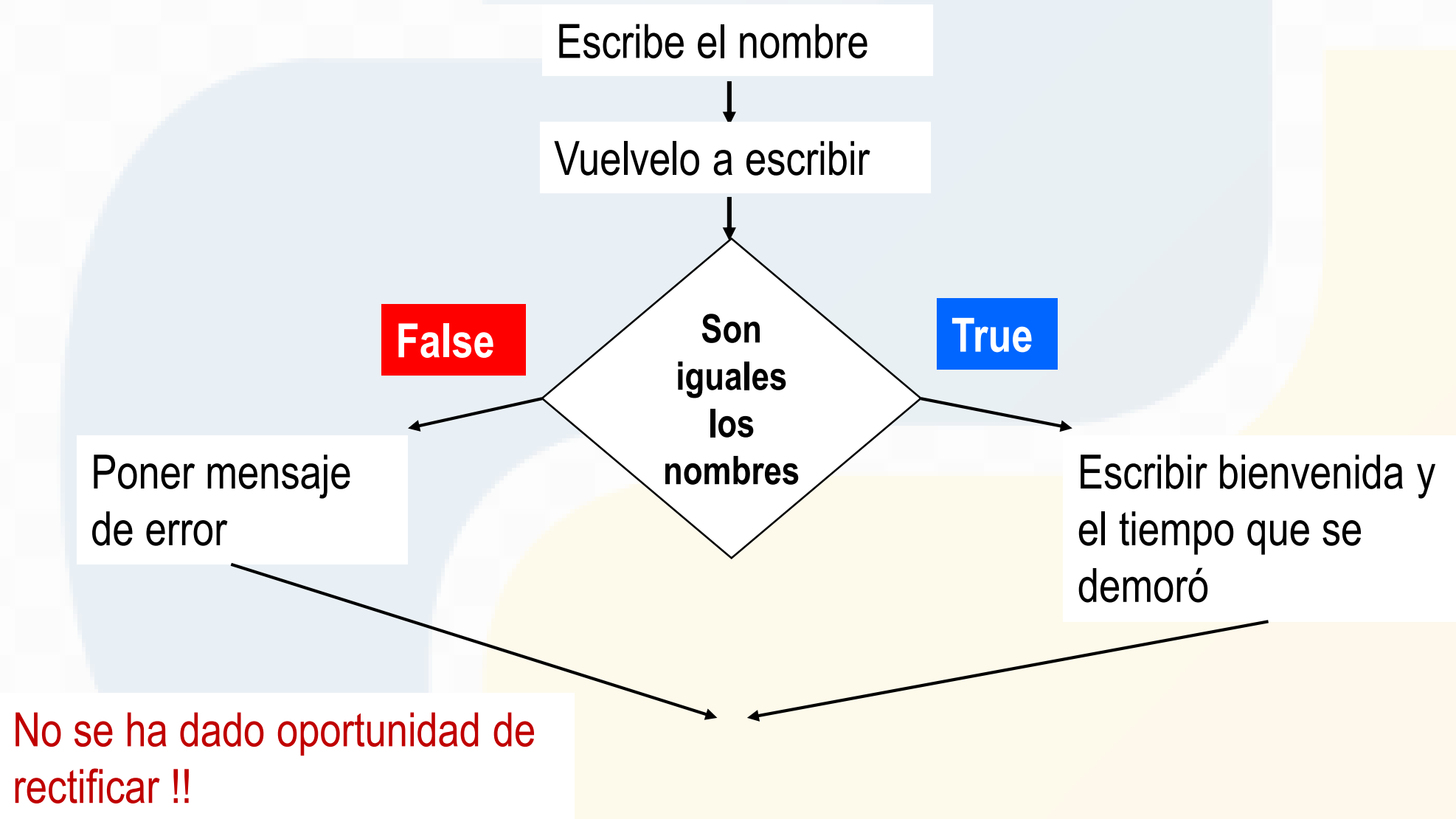
```
if <condicion>:  
    <código>  
    <código>  
    ...  
else:  
    <código>  
    <código>  
    ...  
<código>
```

```
if <condicion>:  
    <código>  
    <código>  
    ...  
elif:  
    <código>  
    <código>  
    ...  
elif:  
    <código>  
    <código>  
    ...  
<código>
```

```
if <condicion>:  
    <código>  
    <código>  
    ...  
elif:  
    <código>  
    <código>  
    ...  
else:  
    <código>  
    <código>  
    ...  
<código>
```

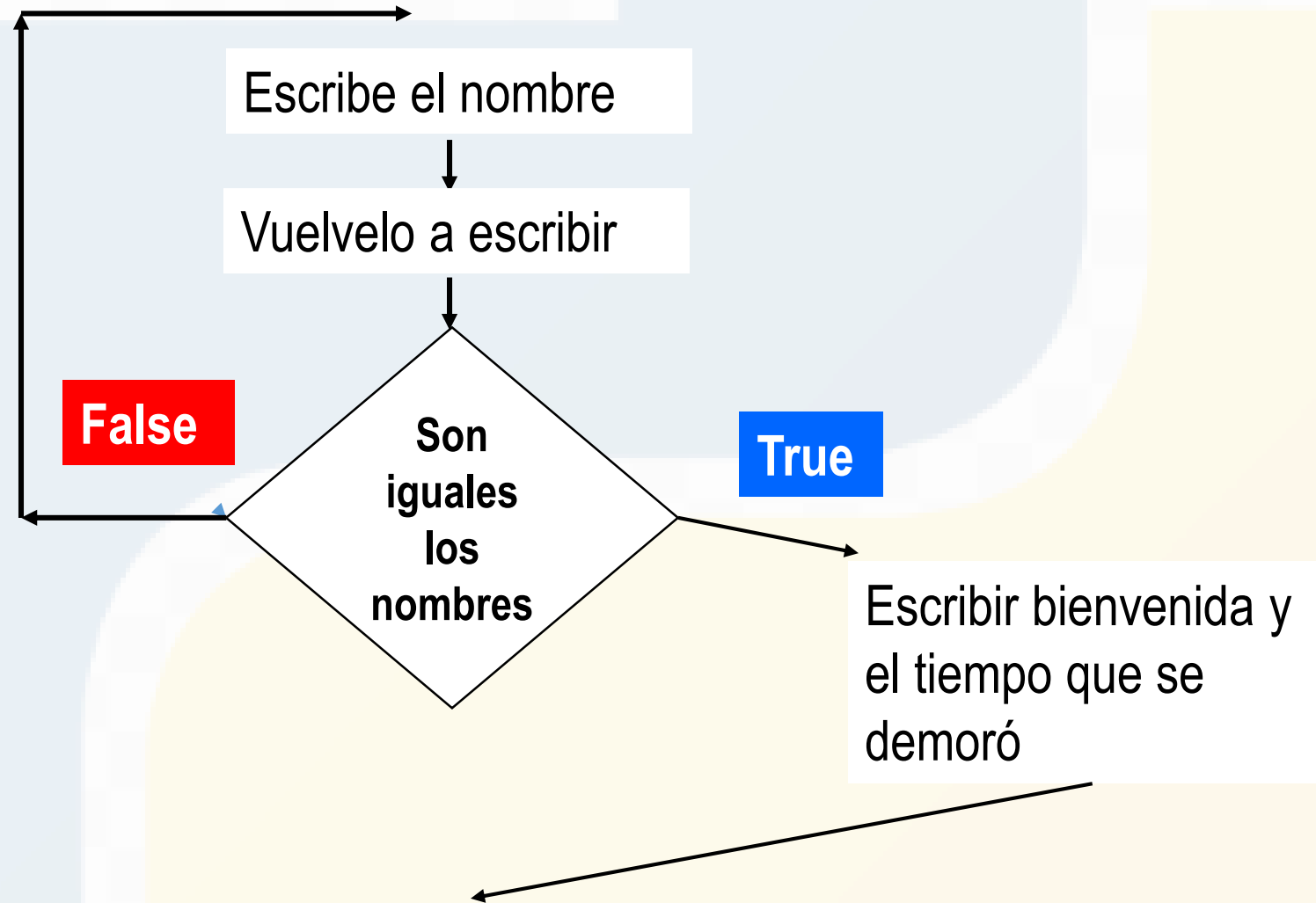
La condición es una expresión que debe evaluar True o False
La sangría es importante

ANTES



CICLOS

Puede rectificar



CICLO WHILE

<código preparación del ciclo>

...

while

<condicion>:

Expresión
Bool

<código>

<código>

...

<código>

Bloque de código que se
ejecuta mientras la
expresión evalúa True

El bloque puede no ejecutarse si desde un inicio la condición evalúa False

El ciclo puede ejecutar indefinidamente si la expresión no dejase de evaluar True

En algún momento dentro del bloque del ciclo debe ejecutarse alguna instrucción que provoque que en la próxima iteración la condición evalúe False

CICLO WHILE

<código preparación del ciclo>

...

while *<condicion>*:

<código>

<código>

...

<código>

Si dentro del bloque se ejecuta una instrucción

break

Se interrumpe el ciclo

La instrucción **break** por lo general será ejecutada a partir de una pregunta condicional **if**

CICLO FOR

<código preparación del ciclo>

...

for <variable> **in** <secuencia de valores>:

<código>

<código>

...

La variable va tomando los valores de la secuencia en cada iteración

La ejecución del bloque termina cuando se han recorrido todos los valores de la secuencia

CICLO FOR con RANGE

```
for <variable> in range(inicio, final, paso):
```

El ciclo comienza con la variable con el valor entero de la expresión *inicio*

El ciclo termina cuando la variable alcanza (o sobrepasa) el valor de la expresión *final* - 1

Antes de repetir cada iteración la variable se incrementa implícitamente en el valor de la expresión *paso*

Si el paso se omite se considera que es 1

EJERCICIOS CP

1. Escriba un código que lea un número entero y calcule el factorial de ese número. Recuerde que el factorial de un número `n` es el producto de todos los números $1*2*3*...*n$. Ejemplo factorial de 4 debe escribir 24
2. Escriba un código que determine si un número que se da como entrada es un número primo. Recuerde que un número primo es aquel que solo es divisible por 1 y por él mismo. Ejemplo 13 es primo pero 49 no lo es porque es divisible por 7
3. Escriba un código que determine si un número es perfecto. Un número es perfecto si es igual a la suma de todos sus divisores sin incluirlo a él. Ejemplo 6 es perfecto porque $6 = 1+2+3$ que son sus divisores
4. Entre los valores de un intervalo, escriba cuáles números en ese intervalo son primos. Note como aquí usará dos ciclos: uno más externo para recorrer los valores del intervalo

EJERCICIOS CP

5. Escriba un código que a partir de un número que lee como entrada escriba cual es el número primo que le sigue
6. **Considere las formas de recorrer los caracteres del ejemplo del Code7. Escriba un código que determine si los caracteres de una cadena forman un palíndromo. Un palíndromo es una cadena que se lee de igual forma de izquierda a derecha que de derecha a izquierda. Por ejemplo oso, larutanatural, dabalearrozalazorraelabad, 51AxyxA15, ??? son palíndromos
7. **Un código como `f = datetime.today()` nos da un valor `datetime` con `.weekday()` del módulo `time` nos da un número entero correspondiente al día de la semana en el que estamos. Dado tres enteros que correspondan al día, mes y año escriba qué día de la semana es.